DOCUMENT RESUME

ED 352 055 IR 054 295

AUTHOR Miley, David W.
TITLE Automating Reference Desk Files with Microcomputers in a Public Library: An Exploration of Data Resources, Methods, and Software.
PUB DATE Dec 91
NOTE 58p.; M.L.S. Thesis, Kent State University.
PUB TYPE Dissertations/Theses - Masters Theses (042)

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS Comparative Analysis; *Computer Software Selection; *Database Management Systems; *Evaluation Criteria; *Information Management; Public Libraries; *Reference Services
IDENTIFIERS Akron Summit County Public Library OH; IBM Personal Computer; *Vendors

ABSTRACT

Many reference librarians still rely on manual searches to access vertical files, ready reference files, and other information stored in card files, drawers, and notebooks scattered around the reference department. Automated access to these materials via microcomputers using database management software may speed up the process. This study focuses on how two divisions of the Main Library of the Akron-Summit County Public Library--Science and Technology, and Philosophy, Religion, and Education--acquire, store, distribute, and use reference information, and examines four software products to determine which one would best meet these needs. Thirteen criteria were established for ranking the software, the most important of which were low-cost, user friendliness, IBM-compatibility, full-text capability, and searching ease. Such a system should be easy to use and low in cost. Four software products were selected and tested: askSam 5.0, Info Select 2.0, FoxPro 2.0, and Q&A 4.0. Test results favored askSam, although automation can be achieved with a number of different products and methods. It is concluded that automating reference files could be a worthwhile project. (Contains 105 references.) (KRN)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED352055

AUTOMATING REFERENCE DESK FILES WITH MICROCOMPUTERS IN A PUBLIC LIBRARY: AN EXPLORATION OF DATA RESOURCES, METHODS, AND SOFTWARE

A Master's Research Paper submitted to the
Kent State University School of Library and Information
Science in partial fulfillment of the requirements
for the degree Master of Library Science

by

David W. Miley

December 1991

BEST COPY AVAILABLE

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

David W. Miley

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

12054295

ABSTRACT

Reference departments have been passed over to a certain degree during the automation of libraries. While the online public-access catalog is available to reference librarians, it is just as much a tool for the patrons.

Reference librarians still rely on manual methods to access vertical files, ready reference files, etc. These information resources are stored in various card files, drawers, notebooks, etc., usually spread about the department.

It may be more efficient to have automated access to these materials via microcomputers using database management software. This paper explores what has been done in this area and also some of the things that can be done.

The file structure, contents, and usage patterns of the reference files of the Akron-Summit County Public Library, Akron, Ohio were studied. System criteria, including low cost, were established for automating the files. Four software products were selected and tested: askSam 5.0, Info Select 2.0, FoxPro 2.0, and Q&A 4.0.

Test results favor askSam, although automation can be achieved with a number of different products and methods. The conclusion is that automating reference files could be a worthwhile project given sufficient funding and resources.

Master's Research Paper by

David W. Miley

B.A., University of Akron, 1990

M.L.S., Kent State University, 1991

Approved by

Adviser Thomas J Froehlich Date 11/19/91

TABLE OF CONTENTS

I. INTRODUCTION 1

II. BACKGROUND 3

III. REFERENCE FILES AND THE DBMS 4

A. FILE STRUCTURES 5

B. USAGE PATTERNS OF THE REFERENCE FILES 8

C. SAMPLE RECORDS 10

IV. DESIRED CHARACTERISTICS OF THE PROPOSED AUTOMATED SYSTEM 11

V. EVALUATING SOFTWARE 12

VI. PROCEDURE FOR EVALUATION 14

A. SOFTWARE SELECTION 14

B. SOFTWARE TESTING 17

1. INSTALLATION 17

2. FAMILIARIZATION AND DOCUMENTATION 18

3. FILE RECORD STRUCTURES AND DATABASE RECORD DESIGN 19

4. TESTING INDIVIDUAL DATABASE PRODUCTS 20

a. ASKSAM 5.0 21

b. FOXPRO 2.0 23

c. INFO SELECT 2.0 25

d. Q&A 4.0 27

C. OTHER DBMS PROGRAMS OF INTEREST BUT NOT TESTED . 30

D. SOME SOFTWARE PROGRAMS ELIMINATED FROM CONSIDERATION 32

VII. CONCLUSIONS AND RECOMMENDATIONS 34

A. SOFTWARE 34

B. OTHER FACTOR TO CONSIDER 37

C. OVERALL RECOMMENDATIONS 38

REFERENCES 39

APPENDIX 42

BIBLIOGRAPHY 43

LIST OF TABLES

Table

Page

1. Criteria for system software 17

2. Evaluation of software tested 29

3. Evaluation of software not selected for testing (based on product reviews and literature) 32

PREFACE

The author became interested in this topic while serving a practicum at the Akron-Summit County Public Library. Having just finished a course in Library Automation, it was somewhat disconcerting to see just how much the reference librarians depended on manual files and systems. It seemed that with some investigation and application of existing technology, the level of patron service might be improved and the librarians' burdens lessened.

All things considered, now that the study is done, available funding may be the deciding factor for implementation. The technology exists to satisfactorily automate reference files. With a little extra effort and money, some of the tasks of the reference librarians could be carried out more efficiently.

The author wishes to thank the Akron-Summit County Public Library, Joyce McKnight, David Jennings, and Kaye Averette for their assistance and the use of the reference files.

I. INTRODUCTION

Reference departments have been passed over to a certain degree during the automation of libraries. While the online public-access catalog (OPAC) is available to reference librarians, it is just as much an end user tool for the patrons.[1]

Reference librarians still rely on manual methods to access vertical files, ready reference files, phone lists, local statistics, community information files, etc. These information resources are stored in various files, drawers, rolodexes, notebooks, etc., usually spread about the department. Some of these files are also indexed manually on cards. For the purposes of this paper, these information resources will be referred to as reference files.

When a patron needs information from one or more of these sources (as determined by the librarian based on the reference interview), the librarian must either have good enough memory and/or enough experience to go directly to the correct reference file, or else sort through a manual index (or indexes) and then go to the file to which the index points. If the information sought is not found in the first file, the librarian must proceed to the next file and repeat the manual search, etc.

It could be much quicker and more efficient to have automated access to these materials via microcomputers using database management software for these reference files. By re-organizing and/or linking computerized reference files and indexes, many or

all of the files could be searched with a single search statement. Thus the librarian could save time and effort in information retrieval. If a terminal were available at each reference desk, it would also save the librarian the effort of moving to each file. (Although it would still be necessary to go vertical files to retrieve folders to which the vertical file index points).

Answers to "repetitive-type questions posed by a fairly predictable patron population" could also be built into these databases, freeing staff for more difficult research.[2] Automation could also permit printing reports for current awareness, bibliographies, and other informational needs.

This paper will explore what has been done in this area and also some of the things that can be done. Various software packages will be tested for suitability using samples of data from the reference files of the Akron-Summit County Public Library (ASCPL), Akron, Ohio. Other software packages will be examined for suitability based on published reviews and product literature.

The paper will focus on librarians as the intended users of the automated reference files. While it may be possible to automate reference files using an OPAC, the proprietary nature of the hardware and software places it outside the context of this paper. It is also outside the scope of this paper to consider the work in artificial intelligence for automated reference service for end users.

II. BACKGROUND

Literature searches turned up some information about using database management software in various ways in libraries. Some sources concentrated on how-to information for specific programs. A few articles described how to choose a suitable database program and offered comparisons of capabilities. There are also numerous reviews of database management systems (DBMSs). The attached bibliography cites many of these works. However, most DBMSs are not really suitable for automating text or reference files.[3, 4]

So far this researcher has not come across literature describing comprehensive solutions for automating reference files. There are bits and pieces of methods that could be implemented haphazardly, but it would require the user to work out many of the details. One article proposed using the search capabilities of a word processing programs for a rudimentary automation system for a vertical file.[5] A few other articles describe local newspaper indexing projects.[6, 7, 8]

Additionally, new versions of software programs come out so frequently that the literature is practically outdated before it is even printed. One company has recently released a product based on askSam 5.0 to automate libraries.[9] One module of the program is advertised as a means to "provide access to the contents of a vertical file ... and ... integrate records of your organization's internal work product (e.g. research)."[10]

Experiments with using expert systems in reference work have been under way for several years. However, the literature still debates whether experts systems have a place in libraries yet.[11, 12, 13] Although the cost of these expert systems in coming down, they still require programming experience for setup and operation.

One article described how teams of graduate students experienced in expert-systems development allotted more than four weeks of a project just to learn how to use the software.[14] Many if not most reference librarians probably do not have that amount of time to devote to learning a software package before beginning a project.

III. REFERENCE FILES AND THE DBMS

In designing a database, one must consider the setting in which it will be used, who the users will be, what types of data it should or could contain now and in the future, and what type of output is desired. One also should study how data is stored, distributed, acquired and used in the present system if one exists.

This study was conducted using the files of two Reference Divisions at the Main Library of the Akron-Summit County Public Library: Science and Technology Division (ST) and the Philosophy, Religion, and Education and Information Division (PRE/ID). Reference divisions at ASCPL are physically separated and

maintain separate reference files. Subject materials contained in each division are determined by Dewey Decimal Classification. Reference questions and material requests by patrons are directed to the appropriate Divisions by subject.

The reference files are used by experienced reference librarians primarily to answer patron queries requiring either ready-reference or non-book printed materials for answers. None of these files are open-access for the public. However, patrons may inspect the materials in the folders of the pamphlet (vertical) files and can examine the club list in PRE/ID after a librarian retrieves it for them. Some of the materials in the pamphlet files do circulate.

Files selected for study from the Science and Technology Division (ST) were the pamphlet, recipe, information, and hazardous waste facilities files. Files sampled from the Philosophy, Religion, and Education and Information Division (PRE/ID) were: pamphlet, speakers bureau, club list, officials (government), and newspapers files.

A. FILE STRUCTURES

The files consist of two basic types: index and full-text. The index files serve as pointers to other files or materials. Examples of these from ST are the pamphlet, hazardous wastes, and recipe files. PRE/ID's pamphlet file is of the same type.

The pamphlet files actually consists of two parts: the actual file cabinets and folders that contain the materials being sought and the index to the files by subject heading that serve as pointers to the file folders. The index files are kept on three-by-five cards at the reference desks in their respective departments. Both the file folders and the index cards are arranged by subject heading.

The index file also contains SEE and SEE ALSO references to direct the librarian to the proper or alternative headings. Headings are derived primarily from Reader's Guide to Periodical Literature with additional specialized or local headings added as needed. Most headings are in natural word order rather than inverted.

ST's hazardous waste facilities file has an index file that consists of three-by-five cards with the names of local sites that have been cited in the literature as containing hazardous wastes. These index cards point to file folders that contain clippings and documents related to these sites and clean-up efforts.

The recipe file in ST consists of three-by-five cards that index the recipes that appear in the Wednesday food section of the Akron Beacon Journal. The individual recipes are indexed by food type, (i.e., desserts). Each card contains several records with the name of the recipe, publication date, and page number. These records point to file folders that contain the actual sections of the newspaper organized by month and year of

publication. The first card in the index drawer lists all the types of foods that the individual recipes are indexed under, serving as a thesaurus. There are no cross references.

The other types of files in use may be considered as forms of full-text files. Examples of these are the information file in ST and the speakers bureau, club list, officials (government), and newspapers files in PRE/ID.

All of these files are stored on three-by-five cards except the club list. The club list has been automated and printed out using PFS: File and is kept in a three-ring notebook that patrons can use for reference. It is arranged alphabetically by the name of the various clubs. The club list contains information that can be used to contact the organizations including officers' names, addresses, and phone numbers. The record structure is similar to what one might use to automate a membership list. Each record also contains a single field for "subject" (the focus of the club). A subject index is included for manual searching. The club list is seldom used in an online mode.

The other files contain information that is needed frequently to answer ready-reference queries or else hard-to-find information that was deemed worthy of recording for future use. Many of the entries consist of only a few words on a single card while others have text that extends over several cards. Filing order varies depending on the content of the file. However, all cards have single headings that they are filed under. SEE and SEE ALSO references may also be found in these files.

B. USAGE PATTERNS OF THE REFERENCE FILES

Usage patterns were observed while serving a practicum at ASCPL and subsequently while employed as a librarian in ST Division. The author also informally interviewed the experienced librarians in both ST and PRE/ID on how, why, and when they used the reference files.

The librarians relied on their experience and judgement on when it would be appropriate to search the reference files. The files were used both as primary and supplementary sources to answer reference questions. They also contain materials used to answer frequent or unusual questions that had been researched previously.

To search the files, the librarian must get up from his or her chair, walk over to the file (or index), and manually search the cards. If the first file is a pointer, the librarian must then walk over to the file cabinets and retrieve the file folder with the proper subject heading and bring it back to the patron (or the phone if the patron is calling in). If the first search was not successful, the librarian must move to the next file or index and perform another manual search, etc., until the material is found or it is concluded the material to answer the question is not contained in the reference files.

The conclusion drawn was that the files might receive more use if access were improved and/or easier and faster. It is not efficient or convenient to go from file to file and drawer to

drawer under the pressure of waiting on a steady stream of patron. If subject headings could be searched at a keyboard, the process could be more productive and make better use of both the human and information resources available.

Instead of having to move from file to file to file, the librarian could perform the searches at the keyboard. If the answer were in a full-text file, the librarian could possibly satisfy the patron's request without having to move from the desk. If the query required materials from the pamphlet file, the librarian could access the subject headings in the index while seated and then go directly to the pamphlet file to retrieve the materials. This could save both time and energy as well as improve the level of reference service.

Another benefit would be to make the reference files more easily accessible to new or substitute librarians in a Division. Currently the Main Library is open for four hours on Sundays during the school year. On these days, it is staffed with a skeleton crew drawn from branches as well as the Main Library. Sometimes the librarians work in Divisions they are not familiar with. In these cases, the learning curve can be rather steep when trying to master the various reference files. With the proper software, it would be possible to view or print a list of all the subject headings in the reference files for scanning or study. Printing a list like this is a task "that would require a prohibitive amount of time in manual systems."[15]

C. SAMPLE RECORDS

In order to determine the characteristics of the files, sample records were typed into a Cambridge Z-88 portable microcomputer and downloaded to a desktop microcomputer for conversion to ASCII text. The file records were subsequently examined for field length, number of fields per record, extensive text within records, and unusual characteristics.

The longest field in the samples is seventy-nine characters long. The most-repeated field in one record is the one for the data under the subject heading of BIRDS -- COLLECTIVE with thirty entries. The former is in the ST pamphlet file while the latter is in the ST Information file. The largest record found fills four index cards, contains 337 words, and converts to an ASCII text-file 2,431 bytes long.

The formats of other files vary widely within the files, notably the PRE club list and speakers files. Part of this variance appears to be due to incomplete information supplied by the organizations and to some fields being pressed into use as comment fields during data entry. The club file sample comes from a printout produced with PFS: File.

The majority of records sampled are short and uncomplicated, however, one does not design the database to fit the short records. It is necessary to search for the largest, most complicated record and design for it as well as the simple ones.[16] An analogy might be that the database record must contain a hole

big enough (and with the right cutouts) for the biggest chunk of data to fit in. If the hole is too small and/or the wrong shape, the data will not fit. But if the hole is too large, there is a tremendous waste of resources.

IV. DESIRED CHARACTERISTICS OF THE PROPOSED AUTOMATED SYSTEM

In order to simplify searching and speed up the process of using the reference files, it would be desirable to search the subject and/or index headings (or other form of main entry) of any or all the reference files in a Division with one search statement. This would eliminate having to switch from file to file. It would also be desirable to search the text in full-text files to increase recall if needed. The software should additionally support full-text retrieval and printing of search results and reports. To aid in determining if a search is too broad or narrow, the system could show the number of retrievals per search. Since usage of each Division's files is confined by subject and data flow to within each Division, there is no incentive or need to be able to combine the reference files from all the Divisions into one all-encompassing file.

The system should also be easy to use. Although ASCPL has an OPAC and a few CD-ROM workstations, personal computer (PC) usage is not widespread among the staff. At the present time there is one personal computer per Division and Branch, with the most sophisticated software in use being PFS: File. Based on

observation and informal interviews, the majority of librarians do not use a PC in the course of their normal duties nor are they comfortable with DOS. Therefore it is important that the system be user-friendly and can be set up and operated without requiring advanced computer skills.

Another consideration is the budget. If there is little funding available for equipment and software, this is a constraining factor. If additional hardware is needed or existing hardware must be upgraded, this must be considered. Cost of the software, technical support, training, upgrades, and consultants (if needed) must be added in. One must also allow for the cost of setup and data entry. Since funding levels are currently frozen at ASCPL, cost is an important factor.

There also must be enough space and electrical outlets to handle additional equipment. Otherwise it may be necessary to buy new furniture and/or remodel. However, this is a hardware factor that must be worked out on a site-by-site basis and is outside the scope of this paper.

V. EVALUATING SOFTWARE

Once it has been determined what the system should ideally provide, it is necessary to study software options for implementation. The software chosen will depend on several factors. Broadly speaking, it must be able to satisfactorily contain and

manipulate the data in the files, be operable by the staff, and fit within budgetary guidelines.

Given the structure of the reference files, a DBMS product would seem to be the logical choice. DBMSs in general handle discrete records and files, and some are capable of handling full-text entries. DBMSs also come in different types (flat-file, relational, etc.) as well as variations on similar types (dBase, XBase). By examining the data structure of the files further, it is possible to better evaluate the DBMSs. It is important to keep in mind that "one of the most important issues of database evaluation is the consideration of trade-offs. No database can perform best on all features... ."[17]

In designing records in a DBMS, one examines the data to be entered for the longest field, the record with the most fields, repeating fields, and any records that have unusual or unique characteristics. Most DBMSs are designed with business applications in mind, i.e., customer lists, inventories, etc. In these databases, the record structure is fairly uniform. However, reference files pose special problems.

In reference files, records commonly have fields that repeat (see, see also, etc.), fields that can vary greatly in length (subject headings, etc.), and lengthy text (information files).[18] If one is working with a DBMS that has fixed-length and single-value fields, this can necessitate record designs that are wasteful of resources. One must design for the largest record in the file and subsequently each smaller record entered will take

up just as much storage space in the computer. This is considered a bad design, but it is better to be liberal with field length than to lack space for an entry at an inopportune time.[19]

There are some ways around fixed-length fields by using memo fields and/or secondary files, but this usually requires extra steps in data entry and retrieval. It may also hamper the ability to format reports. The effort may be more than many users will want to deal with.[20]

The other problem with inflexible DBMSs is that even if one does allow for the largest record currently in the file, a new record encountered in the future could be even larger necessitating a redesign of the record structure. This is not difficult with some DBMS packages, but others may not be so forgiving.[21]

VI. PROCEDURE FOR EVALUATION

A. SOFTWARE SELECTION

Software was selected for testing based largely on how well it met the criteria established for the system overall (see table 1), as well as the ability to handle the structure of the records and files sampled. Software was also selected with an eye to exploring the range of solutions possible with different types of DBMSs. Since the author funded his own project, including buying all the software, it was possible to test only a few selected software packages due to budget limitations.

It had become apparent from reading the literature that most DBMSs would be relatively unsuitable for the task. Reference files demand flexible record structure from a DBMS, a quality not readily found in many programs.[22, 23] Ideally, the DBMS would support variable-length and multiple value fields. Otherwise, space would be wasted in records. However, the costs of the DBMSs that explicitly support these features, Advanced Revelation and INMAGIC, exceed the budget limitations of this project and perhaps the budgets of many reference departments.

Another consideration is ease of use. If the DBMS is too hard to use, it would probably sit idle. Both data entry and searching must be relatively straight-forward, simple, and quick. The ability to import data, i.e., text files, downloads from Dialog, etc., is also desirable. Ease in exporting data as well as formatting and printing reports are other important considerations.

One more factor is cost. With the tight budgets at most libraries, this could be very important. Some DBMSs are quite expensive, while others are fairly reasonable. A few offer run-time or search-only modules at lower cost for use at workstations where files need not be modified. Others offer developers versions that permit distribution of an unlimited number of applications at no extra cost.

Currently, DBMS publishers are fighting for market share and some programs are available at greatly reduced prices. The

industry term for these bargains is competitive upgrade, with purchase at the reduced price usually requiring proof of ownership of a comparable program from another software publisher. While it is possible to save money in this way, one can not depend on these deals being available at the appropriate time or with the right stipulations for eligibility.

It is also important to consider the support policies of the software publisher. Some provide technical support for the cost of a phone call while others require an annual support subscription that may amount to several hundred dollars.[24] Some of the more complicated products may require extensive use of technical support so paying the support fee may be essential for optimum use of the DBMS.[25]

Based on the preceding requirements and information derived from product reviews and literature (see Bibliography), four products were selected as suitable for testing: askSam 5.0, FoxPro 2.0, Info Select 2.0, and Q&A 4.0. These software packages represent four different types of database programs, each with unique features.

Table 1.--Criteria for system software

	must-have	good-to-have	desirable
full-text capability	X		
variable record formats in same file		X	
IBM-compatible	X		
cost within budget	X		
user-friendly (overall)	X		
easy to search	X		
show search hits		X	
free-text entry in any one record			X
record structure easily modified		X	
run-time modules			X
search-only modules		X	
variable-length fields		X	
multiple-value fields			X

B. SOFTWARE TESTING

1. INSTALLATION

For the most part, the software self-installation programs perform smoothly. One simply accepts the default directory or

specifies a different one and answers a few simple questions. FoxPro prompts for how extensive a tutorial is desired. (The full tutorial requires six megabytes of storage on a hard drive).

2. FAMILIARIZATION AND DOCUMENTATION

All of the DBMSs came with both online and text tutorials, some more extensive than others. FoxPro comes with the largest stack of user manuals, nearly eight inches thick. This is appropriate since it is by far the most complex of the four products tested. However, the FoxPro manuals are written at a level intended for an experienced database programmer.

Info Select comes with the slimmest manual, but it is written in language that a beginning to intermediate computer user could understand. The tutorial is brief but demonstrates a wide range of features.

AskSam comes with four manuals that are written for the computer-literate user. The User's Guide includes extensive examples as well as question and answer sections.

Q&A comes with five manuals, but four of them are slim. The User Manual includes detailed explanations as well as numerous screen shots. The majority of the User Manual should be accessible to the beginning computer user while a few sections would be easier for the experienced user to comprehend. The programming functions are printed in the Application Programming Tools

Manual. The entries provide brief explanations, but some examples are given to aid comprehension.

3. FILE RECORD STRUCTURES AND DATABASE RECORD DESIGN

Basically the files can be characterized as two types: index files and information files. The index files serve as pointers, i.e., pamphlet file index, while the information files contain all or at least some of the data on the card/record itself, i.e., speakers file.

Any of these records could be contained by any of the DBMSs selected. The question is how and how well. While a certain design may work, it may not be efficient or permit necessary data manipulation. Another factor to consider is ease of modification and adaptability. For example, some of the records examined have notes penciled in. Ideally, a DBMS should offer the same ease of adding a note to an individual record as writing it with a pencil (metaphorically speaking). If the record is designed with a rigid structure, such additions may be impossible or at least difficult.

Within a division, it may work quite well to relate and/or combine files. Each record could be tagged with the original file name. If desired, a master database of all the subject headings, titles, authority lists, etc. could be distributed to all reference departments within a library for purposes of awareness and sharing of resources.

4. TESTING INDIVIDUAL DATABASE PRODUCTS

It is difficult to compare these DBMSs even when using them to organize the same data because the programs operate in markedly different manners. As an analogy, squirrels are good at climbing trees and ducks are good at crossing ponds. But if the duck has to climb and the squirrel swim, trying to compare their progress would not prove much. Each one is designed for a particular niche. So it is with these DBMSs. Each one produces the results in a different manner.

All four products tested handle the data, searching, and reporting differently. Each has its strengths and weakness and any of them could be pressed into service to do an adequate job with these files. Due to the disparate natures of the DBMSs, a definitive benchmark test could not be devised. However, the ease of working with a DBMS and its flexibility emerged as highly important factors. These factors are somewhat subjective since "ease" and "flexibility" are relative terms. Each product was rated on a one-to-five scale on thirteen factors, with "one" equalling excellent and "five" indicating no implementation, etc. Scores were totalled with low score being best (see table 2).

Part of the goal or this research is to find products that a librarian with intermediate computer literacy could set up and use. If a product is so complex that it requires the services of a programmer and/or consultant, it will not be adaptable to changing reference file needs. Any of these products can be used

by someone familiar with DOS, although they will require a familiarization process with DBMS basics, especially FoxPro. Info Select can be used almost immediately with very little familiarization.

a. ASKSAM 5.0

AskSam is a unique product that allows structured records and free text in the same file. One record can have as much or as little structure as desired and the next record can be entirely different. (However, one must keep in mind the effects on the types of searches and reports one wants when designing records).

AskSam stores information in its files three ways: record mode, document mode, and free mode. A record can contain up to twenty lines of data eighty characters wide. A document consists of linked records with visible boundaries between the records. Free mode is a document with invisible boundaries. Record mode works best for structured data that will fit in a 1600 character space. Document mode works well with longer records that still require boundaries but may contain some free form text. Free mode is best used with documents where boundaries are not important, such as letters, reports, or lengthy text. Free mode permits easy scrolling.

AskSam has three kinds of fields: explicit, implied, and contextual. An explicit field requires a field name followed by a left bracket ([). An implied field is a character or character

string followed by a text string, i.e., $495. Contextual fields are defined by the placement and order of text, i.e., city Kent. It is important to note that askSam does not require the use of any fields whatsoever. Use of them facilitates searching and reporting.

Some of the greatest strengths of askSam are its ease of record design, data entry, and its ability to handle free form text and additional fields at any point. AskSam uses templates that the user designs as the basis for fields for a record. Each file can have up to ten templates. Additionally, new or different fields can be added to any record as needed and fields may be repeated. Field names or other text strings can also be assigned to macro keys for quick and easy use.

Another strength of askSam is its ability to do hypertext and global file searches. One can designate a group of files to be searched, for example, all of the files in a library division. In the hypertext mode, one simply uses the arrow keys to move the highlight over any word or words in a record and askSam will retrieve all other occurrences of that word in the file or files.

AskSam also does basic Boolean searches, but searches the terms strictly left to right with no nesting possible. It will also do proximity searches. Complex searches require formulating search statements. Printing formatted reports requires familiarity with askSam commands, but examples are included in the documentation.

AskSam is a joy to use for setting up records and data entry. There is no need to agonize over record structure or count field lengths to preassign. If an extra field crops up on a record while entering data, one may just type it in directly, even repeated field names. While this practice may not meet the approval of DBMS purists, it seems appropriate for the reference files sampled.

Searching is easy and quick, consisting of merely typing in the keywords or using the hypertext mode. Occasionally the hypertext mode requires use of the built-in editor to trim a line that has pulled in the field name.

Overall, askSam is very flexible and easy to use. Data entry is simple and searching is quick and easy. The only drawback is the need to use command statements for some functions such as reports.

b. FOXPRO 2.0

FoxPro 2.0 is the latest version of a relational (by computer industry standards) DBMS, similar to dBase IV. It can establish a relation between the files provided there is one or more fields in common to link the records.

This relational capability makes it possible to record unique data in one file and repetitive or bulky data in another. This method has been suggested by some as a reasonable course of

action. However, it necessitates extra steps and care in programming and data entry to relate the records.[26]

For example, to store subject headings in one file and SEE and SEE ALSO records in other files, several general steps are required. Each set of records (subject, see, and see also) would need matching unique record numbers to establish relations. This would require extreme care during data entry to avoid lost linkages. It also requires establishment of relations in order to query the databases. This may be more programming than the average librarian wants to do.

One of FoxPro's strengths is variable-length searchable memo fields. This permits storing lengthy entries without wasting space on fields with little or no data. However, these memo fields can not be indexed and it is not simple to search them and display the results.[27]

The memo fields can be displayed in windows that change to match the record that is being browsed in the active database file browse window. But the windows must be sized and positioned carefully to permit viewing of the information. Without a mouse, this could be quite tedious.

Searching indexed text fields requires command line statements and/or use of pull-down query forms. FoxPro will search for text strings with retrieval results that can be puzzling at times. The sections in the manual about searching seem to be written for the experienced DBMS programmer.

Literature accompanying the FoxPro software recommends several programming workshops and how-to books. These would probably be necessary to achieve any great level of success with programming this DBMS for complex tasks.

While FoxPro no doubt works quite well in business and industrial settings in the hands of experienced programmers, it does not seem well suited for inexperienced users trying to automate reference files. It can be pressed into service for this specialized task, but would likely require more effort than many librarians would want to put into using it on a daily basis.

c. INFO SELECT 2.0

Info Select is advertised and reviewed[28] as a Personal Information Manager (PIM), a type of software that helps one organize schedules, reminders, notes, etc. into a personalized database. It also has a "forms" feature which can be used to make formatted records. As with askSam, it is not necessary to use fields since Info Select readily accepts free text. But for ease and accuracy of retrieval it is better to use fields and structured records.

Each record is displayed as a window that may be edited, printed, sorted, or discarded. Text or fields may be added at any place in the record. A window grows in size to accommodate the text it contains.

Records are stored in "stacks," the equivalent to files in other programs. One may load different stacks as needed, but only one stack can be in use at a time. However, stacks can be combined for searches or data manipulation. There is an size limit of ten megabytes per stack.

Searching consists of typing "G" (for the command "Get") and then the text string or words desired. Info Select displays the number of records remaining that match the search terms as each letter is typed in. Results may be displayed at any point in the matching with precision varying accordingly. The search term is highlighted within the retrieved records. Text may also be searched within a given window using the F2 key. Boolean NOT and OR are supported in a convoluted manner.

Info Select is probably weakest in the area of reports. Entire stacks, individual windows, or stacks of windows retrieved by searching can be printed out in the same format as viewed on screen. They can also be output to an ASCII text file for manipulation. But Info Select does not format output.

One specific report that is available is an "overview" which gathers the top line of each window in a stack into one large window (provided one has enough RAM to handle large windows). This might prove useful for printing a listing of all subject headings in a given index file, a task that would be decidedly time consuming with a card index.

In addition to handling reference files, Info Select would be very handy to record tough reference questions and their answers and/or reference desk statistics. Its flexibility would lend itself well to that purpose.

Overall, Info Select is extremely flexible and easy to search, although limited to simple searches. Fields (and/or free text) can be added at any time and their length is not limited. However, the program has limited reporting capabilities.

d. Q&A 4.0

In some ways Q&A is an unlikely candidate for this task. It is the only traditional flat-file DBMS selected for testing. Q&A is included primarily for two features: keyword fields that allow multiple entries, and the Intelligent Assistant,[29] which is an artificial intelligence searching feature. In addition, Q&A supports large fields (32,767 characters) and records (64,512 characters).[30]

With Q&A, one must design a record, including designating field types, placement, and lengths. Fields are fixed length, so significant storage space may be taken up by empty space in small records. Part of this waste is made up for with the multiple-entry keyword fields. These are not multiple-value fields in the sense that the field is repeated as often as needed in each record. Rather, the keyword fields must be designed long enough

to hold all the values, which is still wasteful. The only savings are achieved in simplified search strategies and the reduced number of fields and field names required. The keyword fields will not yield elegant designs, but they work.

The records are easy to design by using arrow keys to place the field names which are simply typed in. Field length is set using visual placement of markers. One then selects the field types to complete the basic record design. Field templates and restrictions on data types can be specified if desired.

The Intelligent Assistant is designed to work with typical database records, i.e. names and numbers, more than textual records. It does a marvelous job with the employee database provided as a tutorial. However, it just got in the road when trying to search the records from the ST pamphlet file. It was much simpler to use the search facilities in the File module of Q&A than the Intelligent Assistant. Nevertheless, the Intelligent Assistant does support Boolean and complex searches and may work better on more complex text records such as downloads from Dialog, etc. with numerous field names to operate on. However, it is overkill for searching the simple records of an index file.

Table 2.--Evaluation of software tested

	askSam	FoxPro 2.0	Info Select	Q&A 4.0
full-text capability	1	2	1	2
multiple record formats in same file	1	5	1	5
IBM-compatible	1	1	1	1
cost within budget	1 $219[a]	4 $499[a]	1 $89[a]	3 $255[a]
user-friendly (overall)	2	4	1	2
easy to search	1	3	1	2
show search hits	1	1	1	5
free-text entry in any one record (non-memo)	1	5	1	5
record structure easily modified	1	2	1	3
run-time modules	2	3	5	5
search-only modules	1	5	5	5
variable-length fields	1	1	1	4
multiple-value fields	5	5	5	5
TOTAL SCORE (LOWEST = BEST)	19	41	25	47

Note: Grading scale is 1-5: 1=excellent, 2=good, 3=fair, 4=poor, 5=does not implement.

[a]Prices are from MicroWarehouse mail order catalog Vol. 3.1 circa September 1991. These are the selling price, not suggested list price.

C. OTHER DBMS PROGRAMS OF INTEREST BUT NOT TESTED

Three other DBMS programs, Advanced Revelation, INMAGIC, and Paradox 3.5, were of interest due to unique features and/or favorable product reviews. However, various factors eliminated them from testing (see table 3).

Two of the programs, Advanced Revelation and INMAGIC, may be appropriate for the tasks of automating reference files. Both programs offer multiple-value and variable-length fields which are highly desirable features for text databases. However, they were not tested due to budget restrictions and the reported complexities of working with these programs.

Reviews emphasize that Advanced Revelation is appropriate for use by program developers only, not end users. One reviewer said it might take three months to program the first project.[31] It is doubtful that many librarians are going to want to devote that kind of time to learning a DBMS. It also is more expensive than FoxPro, dBase IV, or Paradox and offers no educational discounts.

INMAGIC has been reviewed in the library literature as being appropriate for automating an entire library.[32] It can be purchased with modules to do just that. However, it is also rated as difficult to work with.[33] Perhaps one of the major drawbacks may be the publisher's support policy. It costs $300 per year per designated caller for software support. The program itself is also expensive. On top of that, modules for features that

perhaps should be included with INMAGIC are sold as extras for several hundred dollars more.[34]

While both Advanced Revelation and INMAGIC have valuable capabilities, they do not seem to offer ease of use for database design and implementation or use in answering quick reference questions. It may be possible to use these products to automate the entire library and design in the capabilities to automate the reference files. However, it may not be feasible to use them for a stand-alone project for the reference files. The cost and effort would probably be too great.

Paradox 3.5 has received high marks in product reviews.[35, 36] However, it was eliminated from consideration for this type of application because does not implement memo fields. This would make it difficult to handle lengthy (more than 254 characters) text entries. While the task could be accomplished by using repeated fields, it would require more complex programming. For example, to include the longest text field encountered in the files sampled would require ten fields, i.e., TEXT1, TEXT2, TEXT3, ... This could turn into a programming and data-entry quagmire. It is much simpler to use products with memo fields or similar capabilities when dealing with extensive text.

Table 3.--Evaluation of DBMS programs not selected for testing (based on product reviews and literature)

	Advanced Revelation	INMAGIC	Paradox 3.5
full-text capability	1	1	4
variable record formats in same file	5	5	5
IBM-compatible	1	1	1
cost within budget	4 $749[a]	4 $995+[a]	4 $519[a]
user-friendly (overall)	4	4	2
easy to search	[b]	[b]	2
show search hits	[b]	[b]	[b]
free-text entry in any one record (non-memo)	[b]	[b]	5
record structure easily modified	3	[b]	2
run-time modules	5	5	5
search-only modules	5	5	5
variable-length fields	1	1	1
multiple-value fields	1	1	5

Note: Grading scale is 1-5: 1=excellent, 2=good, 3=fair, 4=poor, 5=does not implement.

[a]Prices obtained from various sources circa September 1991. These are approximate selling prices.

[b]Undetermined

D. SOME SOFTWARE PROGRAMS ELIMINATED FROM CONSIDERATION

A few programs were eliminated from consideration due to lack of features, poor reviews, and/or lack of funds for purchase

and testing. This does not mean they can not be used, just that using them could be more difficult or less satisfactory.

DBase IV was not tested due to the lackluster reviews it has received.[35] While it does have variable-length and searchable memo fields, it is not a state-of-the-art DBMS. If one already owns and is familiar with dBase IV, it certainly could be pressed into service. However, there are no compelling reasons to go out and buy a new copy of it.

Some may argue that ZyIndex could be used for automating reference files. ZyIndex is not a DBMS program. Rather it is a text-retrieval program. It is used to index existing text files on computer storage devices and then to search for text strings. The retrieval results can then be viewed and/or pasted into a word processing document, etc. The text files do not have to be structured.

While this type of program may suit some individuals and uses, it does not facilitate printing formatted reports. (Unless one wants to spend time cutting and pasting to format search results). Also, since the text files being searched do not have to be structured, search results could be uneven. But if one already has large quantities of text files stored on computers and does not want to input them into a DBMS, ZyIndex may be a viable alternative.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. SOFTWARE

Which database software program to use depends on many factors: types of data, usage patterns, data flow, budget, level of computer literacy, the task at hand, etc. Even after choosing a DBMS programs, designing databases introduces a wide array of variables.[36] There is no single correct DBMS program or database design. However, some are more appropriate and easier to use than others.

Given the wide array of data and data structures in the reference files sampled (and the even wider array that may exist in other libraries), a DBMS should offer extreme flexibility. Also, given the varying levels of computer literacy probably existent among staff members, the program should be easy to understand and use. Low cost is also an important factor in the case of ASCPL.

The following evaluations are given in order of total score. Of the programs tested, askSam offers the best implementation of these features along with the ability to produce formatted reports (see table 2 for scores). The program will handle text of any length as well as displaying graphics files.[37] Ease of searching is excellent, especially when using the hypertext feature. AskSam can also do global searches on several files at once.

Database design with askSam is easy and forgiving. Fields can be variable length as well as repeated. Extra fields and/or free text can be added or subtracted from any record at any point. By using document or free modes, records of any length can be accommodated. Databases may be indexed if desired, but it is not required. Data entry is simple and direct.

AskSam is also relatively inexpensive with a street price of about $220. Search-only modules are available for substantially less, as well as a developers edition that can be used to design and distribute applications.

Probably the weakest point of askSam is formatting print-outs. But with some study of the command syntax, it is possible to produce the desired output and the commands can be saved as a macro for future use. It is questionable how often one may need to print out extensive reports from the reference files.

The next most flexible and easy to use program is Info Select. Data entry can be nearly as simple as typing if structured records are not required. If structure is needed, record templates can be easily designed and called up for data entry if desired. Additional text or fields can be added to any record as with askSam. Searching is done with a one-keystroke command plus the search string. Info Select may be considered a little strong on recall and a little weak on precision. However, with a slightly more involved search process one can compensate for this.

Info Select could be very useful at the reference desk for quick notes or personalized databases. Each librarian could easily have his or her own disk of information (book selection, serials, bibliographies, etc.) to work on and use while on desk duty. Info Select can easily print out search results or an entire database. However, it is not easy to reformat records for printouts or reports. Info Select is inexpensive with a street price of about $100. It runs as either a memory-resident or stand-alone program.

FoxPro has impressive capabilities. Unfortunately, it takes a programmer to bring them out.[38] The variable-length, searchable memo fields make text retrieval easy in theory. Implementation is another matter. Too many steps are required to search the memo fields easily, quickly, or accurately.

It is also possible to use the relational capabilities of FoxPro to handle repeated fields. But it takes more programming and bother than most librarians may wish to engage in. FoxPro is also the most expensive product tested.

Q&A 4.0 would be fine for simple office automation needs. However, it is out of its element with text databases. Fields are not variable length and the keyword fields are not true multiple-value fields. As a result, unused or partially filled fields would take up valuable storage space. Record design requires careful planning for optimization. Fields will appear in all records in a database and additional fields may not be

added to a single record at a time. Free text must fit within a specified field length.

The Intelligent Assistant seems to be optimized for searching traditional databases, i.e., addresses, inventories, etc. No doubt it is quite useful in business settings, but it did not perform well searching extensive text. In any event, it requires entering a command-like search string. It is probably most useful for simple searches by business people who do not want to take the time to understand how a database works. Basically, all the Intelligent Assistant provides is some automatic substitution of command synonyms.

B. OTHER FACTORS TO CONSIDER

Ideally, each reference desk position would be equipped with a microcomputer for searching the reference files while seated. It may be possible to multi-task and run the OPAC in one window and the reference DBMS in another window. This would depend on proprietary compatibility aspects of the OPAC involved with software such as DESQview or DESQview/X running on the microcomputer. If compatibility issues were worked out, multi-tasking could eliminate the need for placing both a dedicated OPAC terminal and a PC at each position. This would save on equipment costs and desk space. However, evaluation of this aspect is beyond the budget and scope of this paper.

Staff training will also take some time. This may be balanced against the time it takes a newly-hired or substitute librarian to become familiar with the manual files. Automated subject retrieval may prove significantly faster overall than the manual system for familiarization and use of vertical files, etc., even when training time is included.

While some personnel may balk at learning another automation system in addition to the OPAC, others may welcome it, especially new hires who would not be familiar with the local manual system. On the other hand, OPACs have been around for a few years and probably few librarians would choose to go back to card catalogs. Perhaps the best way to win over reluctant staff members would be with demonstrations.

C. OVERALL RECOMMENDATIONS

Automating the reference files could be a worthwhile project with sufficient funds and suitable software and equipment. It will take time and effort to design and implement automated reference file system. However, once it is operating, more time and effort will be saved in return. It could also result in better utilization of information sources and more efficient service.

REFERENCES

1. Harry M. Kriz and Victoria T. Kok, "The Computerized Reference Department: Buying the Future," RQ 25 (Winter 1985): 199.

2. Linda J. Kosmin, "The Electronic Reference Desk: Getting Priorities in Focus," Library Software Review 8 (November-December 1989): 343.

3. Rebecca Price and Christopher Clayton, "Micro DBMS Software: Databases Now Juggle Text, Graphics, Even Music," Government Computer News 9 (September 10, 1990): 53.

4. Ernest Perez, "Managing Text," Data Based Advisor 8 (June 1990): 83.

5. Rachel Applegate, "Wordstar and the Vertical File," Small Computers in Libraries 7 (July-August 1987): 41-2.

6. Geraldine Beare, "Local Newspaper Indexing Projects and Products," The Indexer 16 (October 1989): 227-33.

7. Michael D. Cramer and Mary J. Markland, "Newspaper Indexing with Pro-Cite," College & Research Libraries News 50 (October 1989): 807-8.

8. Charles Anderson, "Getting SAVVY: A Database for Reference Use," Wilson Library Bulletin 61 (September 1986): 41-3, 95.

9. Bruce Flanders, "Dedicated Line," American Libraries 22 (July-August 1991): 630-1.

10. On Point, "TLC Total Library Computerization™," (Washington, D.C.: On Point, 1991). product literature.

11. John Richardson, Jr., "Expert Systems for Reference Service," Bulletin of the American Society for Information Science 14 (August-September 1988): 19-20.

12. John Richardson, Jr., "Toward an Expert System for Reference Service: A Research Agenda for the 1990s," College & Research Libraries 50 (March 1989): 231-247.

13. Mary Micco and Irma Smith, "Expert Systems in Libraries: Do They Have a Place?," Library Software Review 6 (January-February 1987): 25-8.

14. Richard G. Vedder, Maurice G. Fortin, Scott A. Lemmermann, and Ralph N. Johnson, "Five PC-Based Expert Systems for Business Reference: An Evaluation," Information Technology and Libraries 8 (March 1989): 42.

15. Raya Fidel, Database Design for Information Retrieval: A Conceptual Approach (New York: John Wiley, 1987), 13.

16. Jim Seymour, Jim Seymour's PC Productivity Bible (New York: Brady, 1991): 213-21.

17. Fidel, 216.

18. Carol Tenopir and Gerald W. Lundeen, Managing Your Information: How to Design and Create a Textual Database on Your Microcomputer, Applications in Information Management Technology Series, ed. Ching-chih Chen, no. 5 (New York: Neal-Schuman, 1988), 6.

19. Randy Dykhuis, "Database Design," Computers in Libraries 10 (November 1990): 52.

20. Karl Beiser, Essential Guide to dBase III+ in Libraries (Westport, Conn.: Meckler, 1987), 60.

21. Edward Jones, Using dBase IV, (Berkeley: Osborne McGraw-Hill, 1988), 19.

22. Price and Clayton. 53.

23. David Kalman, "15 Relational Databases: Easy Access, Programming Power," PC Magazine 10 (May 28, 1991): 101-200.

24. INMAGIC, "INMAGIC® Price Sheet." (Cambridge, Mass.: INMAGIC, 1991), product literature.

25. Judy Duncan and others, "Multiuser Programmable Databases II," Infoworld 13 (January 14, 1991): 45.

26. Beiser, 172-87.

27. Keith Weiskamp, FoxPro Companion (New York: Brady, 1990): 284-7.

28. George Bond, "Strictly for Personal Information," Byte 15 (September 1990): 197-8.

29. Dykhuis, 51.

30. Symantec Corp., Q&A User Manual (Cupertino, Calif.: Symantec, 1991): "3-4" [section 3, page 4].

31. Duncan, et al., 55.

32. Karl Beiser, "INMAGIC Version 7.2," Computers in Libraries 10 (December 1990): 9.

33. Ibid., 12.

34. INMAGIC Inc., Price List.

35. Kalman, p. 108.

36. Judy Duncan and others, "Multiuser Programmable Databases II," Infoworld 12 (November 12, 1990): 118.

37. Kalman, p. 131-6.

38. Dykhuis, 50.

39. Seaside Software, Inc., askSam User's Guide (Perry, Florida: Seaside Software, 1991): 260-3.

40. Sal Ricciardi, "FoxPro 2.0 Leaves the Rest of the Database Pack in the Dust," PC Magazine 10 (October 15, 1991): 36.

APPENDIX

SOFTWARE PRODUCTS AND PUBLISHERS

Advanced Revelation 2.03
Revelation Technologies Inc.
2 Park Ave.
New York, N.Y. 10016
(800)262-4747

askSam 5.0
askSam Systems
119 S. Washington
Perry, FL 32347-1428
(800)800-1997

DESQview
Quarterdeck Office Systems
150 Pico Blvd.
Santa Monica, CA 90405
(213)392-9701

FoxPro 2.0
Fox Software, Inc.
134 W. South Boundary
Perrysburg, OH 43551
(419)874-0162

Info Select 2.0
Micro Logic Corp.
P.O. Box 174
Hackensack, NJ 07602
(201)342-6518

INMAGIC 7.2
Inmagic Inc.
2067 Massachusetts Ave.
Cambridge, MA 02140
(617)661-8124

Q&A 4.0
Symantec Corp.
10201 Torre Ave.
Cupertino, CA 95014
(408)253-9600

BIBLIOGRAPHY

Acquaviva, James, and Jason Morgan. "Advanced Revelation: A Complete Application Development Environment." Library Software Review 7 (January-February 1988): 46-52.

Anderson, Charles. "Getting SAVVY: A Database for Reference Use." Wilson Library Bulletin 61 (September 1986): 41-3, 95.

Applegate, Rachel. "Wordstar and the Vertical File." Small Computers in Libraries 7 (July-August 1987): 41-2.

Beare, Geraldine. "Local Newspaper Indexing Projects and Products." The Indexer 16 (October 1989): 227-33.

Becker, Karen A. "Getting Documents to the People: Using Finder for an Automated Solution." Illinois Libraries 71 (November 1989): 513-6.

Beiser, Karl. "Database Management Software: A Selection Guide." Wilson Library Bulletin 60 (June 1986): 17-9.

_____. "dBase IV, Version 1.1." Computers in Libraries 9 (October 1990): 32-6.

_____. Essential Guide to dBase III+ in Libraries. Westport, Conn.: Meckler, 1987.

_____. "INMAGIC Version 7.2." Computers in Libraries 10 (December 1990): 9-13.

Bond, George. "Strictly for Personal Information." Byte 15 (September 1990): 197-8.

Booth, Pat F., and M. L. South. Information Filing and Finding. Buckden, Cambridgeshire: ELM Publications, 1982.

Brooks, Terrence A. "Relational Database Design in Information Science Education." Journal of Education for Library and Information Science 26 (Summer 1985): 3-15.

Burton, Paul F., and J. Howard Petrie. The Librarian's Guide to Microcomputers for Information Management. Thetford, England: Van Nostrand Reinhold (UK), 1986.

Capron, H. L. Systems Analysis and Design. Reading, Mass.: Benjamin/Cummings, 1986.

Carleton, Maureen O. and Catherine G. Cheves. "The Vertical File Enters the Electronic Age." Medical Reference Services Quarterly 8 (Winter 1989): 1-10.

Carlson, Rob. "Application Areas for Database Management Packages in Libraries." In SCIL 1986: Software/Computer Conference Proceedings, edited by Nancy Jean Melin, 48-9. Westport, Conn.: Meckler, 1986.

Carter, James W., Daniel R. Masys, David H. Perry, and Dianne E. Tingley. "How to Automate a Service Desk Using dBase III Plus." Online 12 (July 1988): 120-4.

Chong, Alicia. "Topic: A Concept-Based Document Retrieval System." Library Software Review 8 (September-October 1989): 281-4.

Clark, Juleigh. "The Vertical File in South Carolina Academic Libraries." South Carolina Librarian 31 (Spring 1987): 34-6.

Cosgriff, John. "AskSam's Hypertext in the Information Center." Library Software Review 9 (May-June 1990): 158-9.

Cramer, Michael D. and Mary J. Markland. "Newspaper Indexing with Pro-Cite." College & Research Libraries News 50 (October 1989): 807-8.

Dagget, Steve and Sue Dentinger. "Pros and Cons of Run-Time Modules: A Way to Distribute PC-Based Database Applications without a LAN." Library Software Review 9 (May-June 1990): 159-60.

Dejean, David. "Friendlier askSam Version 5.0 Will Still Please Power Users." PC Computing 4 (August 1991): 86.

Duncan, Judy, Zoreh Banapour, Nicholas Petreley, and Linda Slovick. "Data Handlers." Infoworld 12 (November 12, 1990): 115-6, 118, 142, 146-7.

_____. "Multiuser Programmable Databases II." Infoworld 13 (January 14, 1991): 45-7, 50-7, 60.

Dykhuis, Randy. "Database Design." Computers in Libraries 10 (November 1990): 50-3.

_____. "Paradox and the Power of Macros." Computers in Libraries 9 (May 1989): 27-8.

Eddison, Betty. "Computerizing Your Information." Database 9 (April 1986): 85-7.

Enzer, Steven. "Databases: Finding the Right Program for Your Needs." Alternative Software Bulletin 1 (July 1991): 5-9.

Fidel, Raya. Database Design for Information Retrieval: A Conceptual Approach. New York: John Wiley, 1987.

Flanders, Bruce. "Dedicated Line." American Libraries 22 (July-August): 630-1.

Freedman, Barbara. "Solving the Vertical File Buildup." School Library Journal 31 (January 1985): 41.

Gould, Geraldine N., and Ithmer C. Wolfe. How to Organize and Maintain the Library Picture/Pamphlet File. Dobbs Ferry, N.Y.: Oceana, 1968.

Grotta, Daniel. "Second-Tier Software." PC Sources 2 (April 1991): 212-7.

Hane, Paula J. "Database Software: Choose with Care." Database 11 (June 1988): 36-42.

Hayman, Lynne Meyers. "Relational Database File Structure for Serials." Library Software Review 8 (November-December 1989): 335-6.

Higgins, Steve. "AskSam Database Query Tool Delivers Midrange Functions." PC Week 7 (December 3 1990): 59-60.

Hunsberger, Ted. "Using dBase III+ to Index the Local Newspaper." Library Software Review 8 (November-December 1989): 336-7.

Ireland, Norma Olin. The Pamphlet File. Boston: F.W. Faxon, 1954.

Jones, Edward. Using dBase IV. Berkeley: Osborne McGraw-Hill, 1988.

Kalman, David. "15 Relational Databases: Easy Access, Programming Power." PC Magazine 10 (May 28, 1991): 101-200.

Kosmin, Linda J. "The Electronic Reference Desk: Getting Priorities in Focus." Library Software Review 8 (November-December 1989): 342-3.

Kriz, Harry M. and Victoria T. Kok. "The Computerized Reference Department: Buying the Future." RQ 25 (Winter 1985): 198-203.

Lazinger, Boaz. "An Integrated Database/Word Processing System for Reference Document Retrieval Using dBase IV." In Computers in Libraries '90: Proceedings of the 5th Annual Computers in Libraries Conference, edited by Nancy Melin Nelson, 103-4. Westport, Conn.: Meckler, 1990.

Lazinger, Susan S. "*Producing an LCSH Authority List for Special* Libraries with dBase." Electronic Library 8 (February 1990): 8-14.

Lee, Frank. "Dead End for Documents -- Alternatives to the Vertical File." Public Library Quarterly 6 (Fall 1985): 51-5.

Lundeen, Gerald. "Software for Managing Personal Files." Database 12 (June 1989): 36-48.

Marcum, Louise. "The Vertical File -- A Necessary Tool." South Carolina Librarian 31 (Spring 1987): 37-8.

Marks, Kenneth. "AskSam: An On-line Personnel System." In Computers in Libraries 1989: Conference Proceedings, edited by Nancy Melin Nelson, 91-3. Westport, Conn.: Meckler, 1989.

Marshak, Ronni T. "Managing Information." Patricia Seybold's Office Computing Report 13 (August 1990): 2.

Marshall, Patrick. "AskSam 5.0 Adds Databaselike Relationality." Infoworld 13 (August 12, 1991): 67-71.

Marshall, Patrick, Rod Chapin, and Patrick Lyons. "Combing the Stacks." Infoworld 12 (December 24/31 1990): 31-43.

Martin, James. "1990s Increase Demand for Text-Query Improve ments." PC Week 7 (October 1 1990): 64.

Mater, Dee A. "Pro-Cite: Developing a Statistical Resources Database." Library Software Review 9 (January-February 1990): 25-6.

Micco, Mary, and Irma Smith. "Expert Systems in Libraries: Do They Have a Place?" Library Software Review 6 (January-February 1987): 25-8.

Micco, Mary, Irma Smith, Su-ann Hsiao, and Siriat Intaravitak. "Knowledge Representation: Subject Analysis." Library Software Review 6 (March-April 1987): 82-7.

Micro Logic Corp. Info Select: Operations Guide. Hackensack: Micro Logic Corp., 1990.

Miller, Shirley. The Vertical File and its Satellites. Littleton, Colo.: Libraries Unlimited, 1979.

Myrvold, Barbara, and Cheryl Buchwald. "PC-File and Authex: Creating a Local History Subject Information File." Library Software Review 9 (January-February 1990): 27.

Nieuwenhuysen, Paul. "A Bibliography of Text Information Management Software for IBM Microcomputers and Compatibles." Electronic Library 8 (August 1990): 254-78.

O'Malley, Christopher. "dBase One-Upmanship." PC Sources 1 (December 1990): 85-8.

Oppenheim, Pam. "The Latest Word on Text-Based Management Systems." National Productivity Review 9 (Winter 1989/90): 83-8.

Otten, K. W. "Information Tools: Hardware and Software Ergonomics Determine User Acceptance and Productivity." Journal of Information and Image Management 17, no. 4 (1984): 19-26.

Pao, Miranda Lee. Concepts of Information Retrieval. Englewood, Colo.: Libraries Unlimited, 1989.

PC Sources. "Advanced Revelation 2.0" 2 (March 1991): 458.

PC Sources. "Stalking the Right Database Manager." 2 (March 1991): 431-41.

Perez, Ernest. "Managing Text." Data Based Advisor 8 (June 1990): 83-93.

Phalen, Heather. "Supplementary Resources for Enhancing Reference Service." Ohio Library Association Bulletin 55 (July 1985): 17-9.

Portante, Tom. "We're Solving the Wrong Problems." Patricia Seybold's Office Computing Report 13 (August 1990): 15-8.

Price, Rebecca, and Christopher Clayton. "Micro DBMS Software: Databases Now Juggle Text, Graphics, Even Music." Government Computer News 9 (September 10, 1990): 50-3.

Purcell, Royal. "Computers and Reference Service." Library Software Review 8 (September-October 1989): 256-64.

Reid, Clifford. "Comparing Text, Document, and Relational Database Management Systems." Library Software Review 9 (March-April 1990): 80-2.

Rholes, Julia M., and Suzanne D. Gyeszly. "Pro-Cite as a Management Tool for a Travel and Tourism Collection." Collection Building 10, no. 3-4 (1989): 47-50.

Ricciardi, Sal. "FoxPro 2.0 Leaves the Rest of the Database Pack in the Dust." PC Magazine 10 (October 15, 1991): 35-6.

Richardson, John, Jr., "Expert Systems for Reference Service." Bulletin of the American Society for Information Science 14 (August-September 1988): 19-20.

_____. "Toward an Expert System for Reference Service: A Research Agenda for the 1990s." College & Research Libraries 50 (March 1989): 231-247.

Roberts-Buchanan, Frances, and Judy K. Rule. "Sidebar 3: The Statewide Information and Referral Service at Cabell County Public Library." Library Hi Tech 7, no. 1 : 26-8.

Ross, Linda N. "PC-Info: The Information and Referral Software for Microcomputers." Reference Service Review 15 (Winter 1987): 21-34.

Rowley, J. E. "Guidelines on the Evaluation and Selection of Library Software Packages." Aslib Proceedings 42 (September 1990): 225-35.

Seaside Software. askSam User's Guide. Perry, Florida: Seaside Software, 1991.

Seymour, Jim. Jim Seymour's PC Productivity Bible. New York: Brady, 1991.

Shaw, Richard Hale. "Flat-File Databases: Fast Answers to Easy Questions." PC Magazine 9 (June 26, 1990): 245-65.

Sherer, Paul M. "AskSam Gains Relational Database Features." PC Week 7 (October 22 1990): 77.

Smith, Dana E. "QPROMPT: The Reference and Information Authoring Program." Small Computers in Libraries 7 (November 1987): 24-6.

_____. "The Information Station." Small Computers in Libraries 7 (June 1987): 26-7.

Special Libraries Association, Newspaper Division. Automating the Newspaper Clipping File: A Practical Guide. Washington, D.C.: Special Libraries Association, 1987.

Spector, Lincoln. "A Livelier Q&A." PC World 9 (July 1991): 103-5.

Still, Julie. "The Vertical File in Academic Libraries." Show-Me Libraries 39 (Spring 1988): 36-8.

Strassler, George. "Don't Pass Up the Librarians." Nebraska Library Association Quarterly 20 (Summer 1989): 38-42.

Symantec Corp. Q&A User Manual. Cupertino, Calif.: Symantec Corp., 1991.

Szynaka, Edward M., and Anne Henley Cain. "Local Databases: The Future of Public Libraries." Journal of Library Administration 11, no. 1/2 (1989): 189-98.

Tenopir, Carol. "Software Options for In-House Bibliographic Databases." Library Journal 112 (May 15, 1987): 54-5.

Tenopir, Carol, and Gerald W. Lundeen. Managing Your Information: How to Design and Create a Textual Database on Your Microcomputer. Applications in Information Management Technology Series, ed. Ching-chih Chen, no. 5. New York: Neal-Schuman, 1988.

_____. "Software Choices for In-House Databases." Database 11 (June 1988): 34-6.

Toronto Public Libraries. Subject Headings for Vertical Files. Toronto: Toronto Public Libraries, 1964.

Valerie J. Nurcome. Review of The Essential Guide to dBase III+ in Libraries, by Karl Beiser. In Library Review 38, no. 3 (1989): 70.

Vedder, Richard G., Maurice G. Fortin, Scott A. Lemmermann, and Ralph N. Johnson. "Five PC-Based Expert Systems for Business Reference: An Evaluation." Information Technology and Libraries 8 (March 1989): 42-54.

Verrier, Monique. "AskSam Takes Instruction Well." Data Based Advisor 8 (September 1990): 70.

Weiskamp, Keith. FoxPro Companion. New York: Brady, 1990.

Welsch, Erwin K. "Information Management with askSam and Memory Mate." Computers in Libraries 9 (April 1989): 28-9.

White, Ron, Paul Bonner, Kathleen Richards, Peggy Wallace, Dale Lewallen, Randy Ross, and Don Steinberg. "Confine & Conquer Information Overload." PC Computing 3 (August 1990): 70-86.

Wilson, Tom. "AskSam." Electronic Library 8 (August 1990): 282-5.

Yerkey, A. Neil. "Enhanced Memo Field Processing in dBase IV." Computers in Libraries 10 (February 1990): 35-6.

_____. "Password Protection for dBase Applications." Microcomputers for Information Management 6 (March 1989): 33-45.

Young, Heartsill, ed. The ALA Glossary of Library and Information Science. Chicago: American Library Association, 1983.